

What is CDAT?

A brief tour

What is CDAT?

- A quick tour of CDAT, showing:
 - VCDAT – the CDAT GUI
 - Running CDAT from Python scripts
 - Running CDAT interactively
 - Applications on top of CDAT
 - Quick look at some code
 - Documentation

CDAT Propaganda

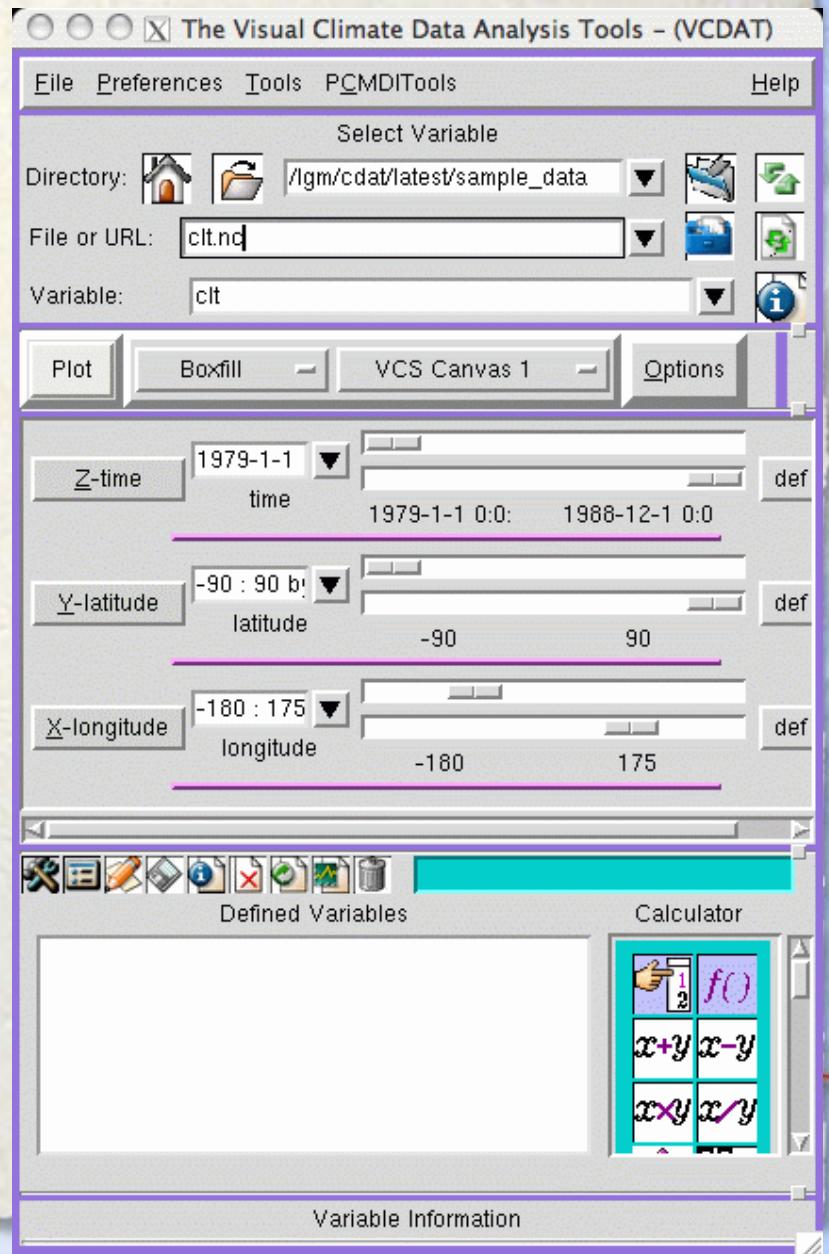
Introducing CDAT...

- Designed for climate science data.
- Scriptable directly from the Python interface.
- Analysis, conversion, sub-setting and array operations.
- Interfaces to Fortran and C.
- Visualization and Control System (VCS).
- Graphical User Interface (GUI) - Visual CDAT (VCDAT).
- XML representation (CDML) for datasets.
- Open-source and free.
- Integrated with other packages such as the Live Access Server (LAS).



VCDAT – Visual CDAT

- VCDAT lets you get familiar with many parts of CDAT without actually knowing any python, it will also generate **“commented”** python scripts for you.
- Start by typing “vcdat” at the command line.



VCDAT – Visual CDAT

- VCDAT is usually the first tool that newcomers to CDAT play with.
- It provides a Graphical User Interface (GUI) to CDAT's functionality.
- Advantages:
 - intuitive interface for quick start
 - no need to learn scripting language (python)
 - binds different CDAT sub-packages seamlessly (*cdms*, *vcs*, *cdutil*, *cdtime* etc.)
 - provides tips on how to script CDAT.

VCDAT is built in Python using the **Tkinter** and **Python MegaWidgets (PMW)** libraries.



File Options Tools PCMDITools

Help

VCDAT – Visual CDAT

Directory: /home/tornado/astephen/cdat/pres_demo



File: Type in a file name or select file using pull down arrow -->.



Variable: Type in variable name or select variable using pull down arrow -->.



Plot

Boxfill

VCS Canvas 1

Options

Define

File Select

Directory:

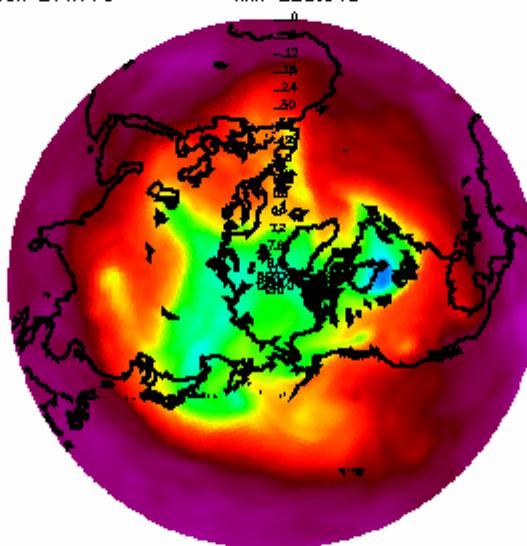
data.xml
 gpas1999010100tcc.ctl
 gpas1999010106tcc.ctl
 gpas1999010112tcc.ctl
 gpas1999010118tcc.ctl
 mydata.nc

File name:

Files of type:

1. - Visualization and Control System (VCS)t Temperature (K)
Mean 255.729 Max 271.779

Min 225.045



224	228	232	236	240	244	248	252	256	260	264	268	272
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



Running CDAT from Python scripts

- CDAT IS Python!
- You can combine it with any python code.
- Python interfaces to Fortran/C/C++ allow you to bind to lower level languages.
- Python is really useful for general scripting.
- CDAT's OOP model allows flexible control of data objects.
- You can build packages directly on top of CDAT since it is python.

Running CDAT from Python scripts

```
#!/usr/bin/env python

print "I am a python script."

print "Let's import some CDAT modules..."

import cdms, vcs

print "Open a data file, grab some data..."

f=cdms.open("myfile.nc")

var=f('temperature', latitude=(0,90),
      time="2004-12-17")

print "Plot the data..."

canvas=vcs.init()

canvas.plot(var)

print "So long!"
```

Running CDAT interactively

- You can work interactively with CDAT because python has an interactive prompt.
- Run ‘idle’ or ‘python’:

```
>>> print "hello"  
>>> import cdms, Numeric  
>>> arr=Numeric.array([1,3,4,6], 'f')  
>>> lon=cdms.createAxis([0.,0.1,0.2,0.3])  
>>> lon.id="longitude"  
>>> lon.designateLongitude()  
>>> lon.units="degrees_east"  
>>> lon.standard_name="longitude"  
>>> var=cdms.createVariable(arr, axes=[lon])  
>>> print var
```

Applications on top of CDAT

- Building on top of CDAT is simple, some example applications are:
 - NetCDF CF-convention checker:
<http://titania.badc.rl.ac.uk/cgi-bin/cf-checker.pl>
 - BADC Data Extractor:
<http://cdat.badc.nerc.ac.uk/cgi-bin/dxui.py>

NetCDF CF-convention checker

The screenshot shows a web page titled "CF-Convention compliance checker for NetCDF format". At the top, there is a blue header bar with navigation links: Home, My BADC, Data, Search, Community, and Help!. Below the header, the main title is displayed in a large, bold, dark blue font. A descriptive text follows, explaining the purpose of the checker and crediting Rosalyn Hatcher and the Met Office. It also mentions that it is a Beta release and invites suggestions for improvement via email. Further down, instructions for file selection are given, along with links to the NetCDF format and CF Convention pages. At the bottom, there is a file input field, a "Browse..." button, and a "Check file" button. The footer contains links to "Home" and "Contact", and a timestamp indicating the page was last modified on 12/06/2004 at 21:41:50.

CF-Convention compliance checker for NetCDF format

This form allows you to run the 'cfchecks.py' script to check that the contents of a NetCDF file comply with the **Climate and Forecasts (CF) Metadata Convention**. The CF-checker was written by Rosalyn Hatcher of the Hadley Centre for Climate Prediction and Research, UK Met Office. This work was supported by PRISM (PPrograme for Integrated Earth System Modelling).

The CF-checker is a Beta release and development work is currently underway. If you have suggestions for improvement then please e-mail Rosalyn Hatcher (rosalyn.hatcher@metoffice.com).

To check your file, please enter the file name in the box below then press *Check file*. You can use the *Browse...* button to help you select the file.

For details of the NetCDF format and the CF Convention go to the [NetCDF format](#) page or the [CF Convention](#) page.

File:

[Home](#) [Contact](#) Last Modified: 12/06/2004 21:41:50

BADC Data Extractor

0.0 W 357.5 E
-90.0 S

Select from map Note that the map Java applet ma

[Note about interpolation methods.]

Vertical Domain

Levels
Single level

Time

Dataset 1: Start time
1979 01 01 00 00 00
year month day hour min sec
1979 01 01 00 00 00
End time

Format
NetCDF Note that you should choose NetCDF format if

Proceed

http://cdat.badc.nerc.ac.uk/dx_extra/LiveMap_30/extractorMap.html - Microsoft ...

Choose this selection

VIEW: Longitude-Latitude

90.0 N 180.0 W 180.0 E

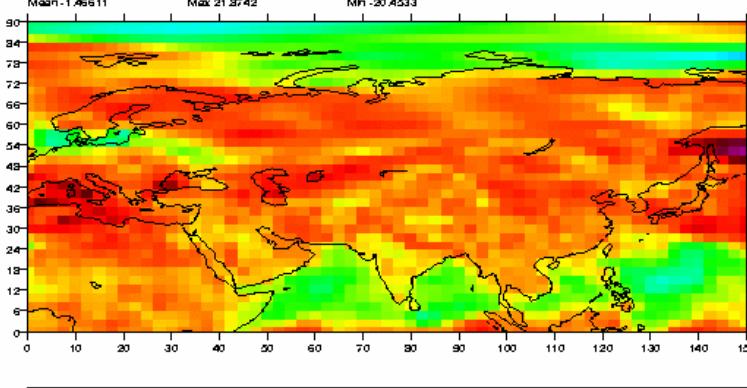
Projection: Standard
Dimensions: 600x400

Plotting your data...

Request processed...

NERC Centres for Atmospheric Science
NATIONAL ENVIRONMENT RESEARCH COUNCIL

Source: British Atmospheric Data Centre (<http://badc.nerc.ac.uk>)
new_year Difference dataset: no10u - no10u
Mean-1.49511 Max 21.3742 Min -20.4533
1979/1/1 0:0:00



Some example code (1)

- calculate a long-term average:

```
import cdms, cdutil  
f=cdms.open('decades_of_data.xml')  
wspd=f('wind_speed', time=("1980-01-01",  
                           "1999-12-31")  
mean_wspd=cdutil.averager(wspd, axis="t")
```

- define wind-speed from u- and v-components:

```
import cdms  
f=cdms.open('met_variables.xml')  
u=f('u_wind')  
v=f('v_wind')  
wspd=(u**2+v**2)**0.5
```

Some example code (2)

- subset a dataset, selecting a spatiotemporal region

```
import cdms
f=cdms.open('4d_dataset.ctl')
print f.variables
a=f('o3mr', time=slice(3,5), lon=(10,100),
     lat=(30, 60), level=1000)
b=a(lon=50, lat=(40,50))
```

- aggregate 1000s of files into a small XML file.

```
$ cdscan -x my_dataset.xml /my_data/*/*.nc
$ python
>>> import cdms
>>> f=cdms.open('my_dataset.xml')
>>> print f.variables # prints a variable list
>>> var=f["temp"]
```

Plotting some data

```
astephen@westerly$ cp /badc/ecmwf-e40/.cache_gp/gp/as/1999/01/01/gpas19990101??tcc.* ./
astephen@westerly$ n
Last login: Tue Feb 24 15:10:02 2004 from westerly.badc.rl.ac.uk
astephen@neptune$ cd cdat/pres_demo/
astephen@neptune$ ls
gpas1999010100tcc.ctl      gpa
gpas1999010100tcc.grb      gpa
gpas1999010100tcc.grb.idx  gpa
gpas1999010106tcc.ctl      gpa
gpas1999010106tcc.ctl      gpa
```

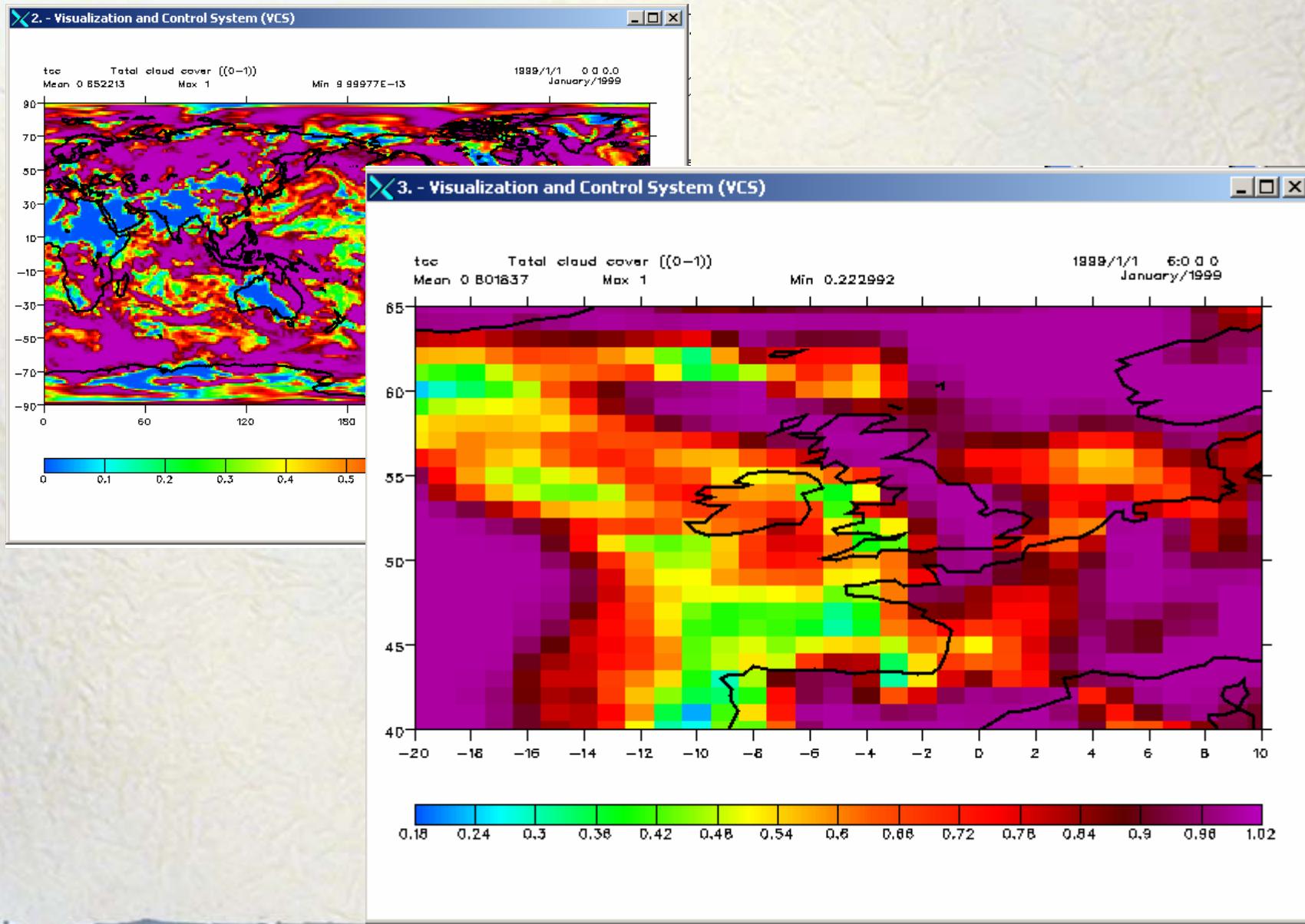
```
cdscan -i 6 -x data.xml gpas19990101*ctl
```

```
astephen@neptune$ cdscan -i 6 -x data.xml gpas19990101*ctl
Finding common directory ...
Common directory:
Scanning files ...
gpas1999010100tcc.ctl
Setting reference time units to hours since 1999-1-1 0:0
gpas1999010106tcc.ctl
gpas1999010112tcc.ctl
gpas1999010118tcc.ctl
data.xml written
```

```
astephen@neptune$ cdat
Executing /usr/local/cdat/bin/python
Python 2.2.1 (#1, Apr 25 2002, 10:02:23)
[GCC 2.96 20000731 (Red Hat Linux 7.1.14-1)]
Type "help", "copyright", "credits" or "license" for more information.
Importing Ag's startup script: .
>>> import cdms
>>> import vcs
>>> f=cdms.open('data.xml')
>>> f.variables
{'tcc': <Variable: tcc, dataset: None>}
>>> f.getVariables()
[<Variable: tcc, dataset: None>]
>>> tcc=f('tcc')
>>> canvas=vcs.init()
>>> canvas.plot(tcc)
<vcs.displayplot.Dp instance at 0x2a0d00>
>>> subset=tcc(time="1999-01-01")
>>> canvas2=vcs.init()
>>> canvas2.plot(subset)
<vcs.displayplot.Dp instance at 0x2a1000>
```

```
import cdms , vcs
f=cdms.open('data.xml')
f.variables
tcc=f('tcc')
canvas=vcs.init()
canvas.plot(tcc)
canvas2=vcs.init()
subset=tcc(time="1999-01-01 06:00")
canvas2.plot(subset)
```

Plotting some data



CDAT Documentation

- You can find online documentation at <http://cdat.sf.net> on:
 - CDMS
 - Numeric
 - VCS
 - CDAT
 - Tutorials (download)
 - Movie tutorials

See also the *links and support* presentation provided.